

725

PRESENTATION 3.1.6

**N91 - 17038**

**ADVANCED SOFTWARE INTEGRATION**



October 25, 1989

Abstract - Space Transportation Avionics Technology Symposium

November 7-9, 1989

John R. Garman, Johnson Space Center

**ADVANCED SOFTWARE INTEGRATION - THE CASE FOR ITV FACILITIES**

Avionics software development has enjoyed an incredible evolution during the last 20 years. From Apollo, through Shuttle, and into the current plans for Space Station - the array of technologies and methodologies involved in the development and integration of avionics software has moved almost as rapidly as computer technology itself.

Future, near future, avionics systems involve major advances and risks in the following areas:

- a) Complexity  
( *technology, functionality* )
- b) Connectivity  
( *distributed, networks, remote resources* )
- c) Security  
( *privacy, protection, integrity in development and maintenance* )
- d) Duration  
( *" never ending" and evolutionary* )
- e) Software Engineering  
( *layers, encapsulation, objects, etc.* )

From an architectural point of view, the systems will be much more distributed (including flight/ground), involve "session"-based user interfaces, and have the layered architectures typified in the "layers of abstraction" concepts popular in networking (e.g. OSI) and software engineering design standards today.

Perhaps most important, and typified in the NASA Space Station Freedom program, will be the highly distributed nature of software development itself. Whether it be the integration of "off-the-shelf" or reusable products, or the integration of

components separately developed by teams of contractors and subcontractors distributed to remote locations, it is the "decentralization" of software development itself that probably contributes the most fundamental changes in avionics software management and integration in the 90's.

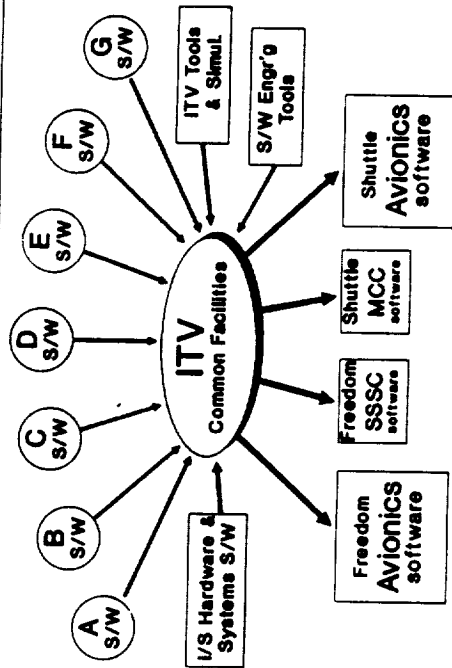
Systems composed of independent components developed in parallel must be bound by rigid standards and interfaces, the clean requirements and specifications. Nonetheless, it is the integration of the separate components into whole which provides the real challenge. Avionics software provides a compounding challenge in that it can not be "flight tested" until the first time it literally flies. This normally means that man-rated or safety critical avionics software must obtain that rating and certification in simulated environments of the real systems and vehicles. It is this combination of verification in a "virtual" target environment coupled with the distributed nature of the component development, which led to special ITV (Integration, Test, and Verification) concepts for the Shuttle, and now Space Station Freedom Programs. The latter employs a "Multi-System Integration Facility" concept for its avionics and ground mission systems. While the name and scope has and will evolve, the underlying concepts, vis a vis software integration remain the same.

It is the binding of requirements for such an integration environment into the advances and risks of future avionics systems themselves, enumerated above, that form the basis of this paper and the basic ITV concept within the "never-ending" development and integration life cycle of Space Station Mission and Avionics systems.

PRECEDING PAGE BLANK NOT FILMED

# Advanced Software Integration

SPACE TRANSPORTATION AVIONICS TECHNOLOGY SYMPOSIUM  
ADVANCED SOFTWARE INTEGRATION



STARS01-1/91031

## Major Objectives

- Maintaining reliability in increasingly complex software and information systems (contrasts in STS and SSFP avionics).
- Enabling evolution (functionality, technology, connectiveness) in systems which are now "never-ending".
- Managing increasingly distributed work-packets and efforts in the development of applications software for the advanced systems.
- Reuse and commonality (across systems and programs) both an operations efficiency (training, management, etc.) and as a productivity item.

STARS02-1/91031

## Key Contacts & Facilities

### Contacts:

John R. Garman/JSC(FA)  
Ed Chevers/JSC(FR)  
Rick Coblenz/JSC(FR)  
Jack Seyl/JSC(FS)  
Charles McKay/UHCL (JSC)

### Facilities:

Information Systems Technology Lab (ISTL)/JSC(FA)  
Avionics Development Lab (SSFP/WP2)/JSC(FR)  
Software Development Facility (NSTS)/JSC(FR)  
Support Software Environment Development Facility (SSEDF)/JSC(FR)  
Mission Systems ITV Facility/JSC(FS)

## Major Milestones (1990-1995)

- Operation of STS SAIL and SPF (1976)
- SSE Baseline for SSFP (1990)
- Avionics Integration and ITV baseline for SSFP (1990)
- Mission Systems ITV (MSITV) Facility Design (1990)
- ADF and MSITV FOC (1992)
- Shuttle and SSFP ITV commonality (1997)

# Advanced Software Integration (cont'd)

## Technology Issues

- Containment of growing drivers: complexity, connectivity, security, and architectures
- Standardization of I/S "layers" - industry standards
- Virtual target environments (exact simulation of target platform allowing diagnostics)
- "Project Object Database" - the database and management technologies involved in creating a single unambiguous image of the entire distributed software system
- Integration of heterogeneous products designed against common standards (both the host and target domains)
- Software Lifecycles modeled against evolutionary development and maintenance (vs. waterfall)

SLIDE 70101

## Major Accomplishments/"Inabilities"

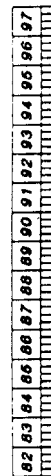
- ✓ Major Accomplishments
  - Establishment of RICIS
  - Establishment of SSE development effort
  - Baselining of commonality in applications tools and UI for SSFP
  - Industry evolution toward standardization of I/S layers
- ✓ Major "Inabilities"
  - Duplication of effort across Programs/Projects
  - Proliferation of mission supporting software
  - Inability to fully utilize COTS
  - Inability to upgrade existing capabilities

## Candidate Programs

- NSTS Avionics Flight Software
- NSTS Mission Control Center Upgrade
- NSTS Other I/S
- SSFP Data Management System (avionics)
- SSFP Mission Control and Trainers
- SSFP Other I/S
- Advanced Programs (Lunar/Mars)

SLIDE 70102

## SIGNIFICANT MILESTONES Advanced Software Integration



R&T

GSFC SEL, JSC RICIS, CMU SET

Adv. Level

SSFP SSE, JSC ISTL

DDT&E

ADF, MSITV (SAIL-237Z)

Projected Level 6 Tech. Maturity ▼

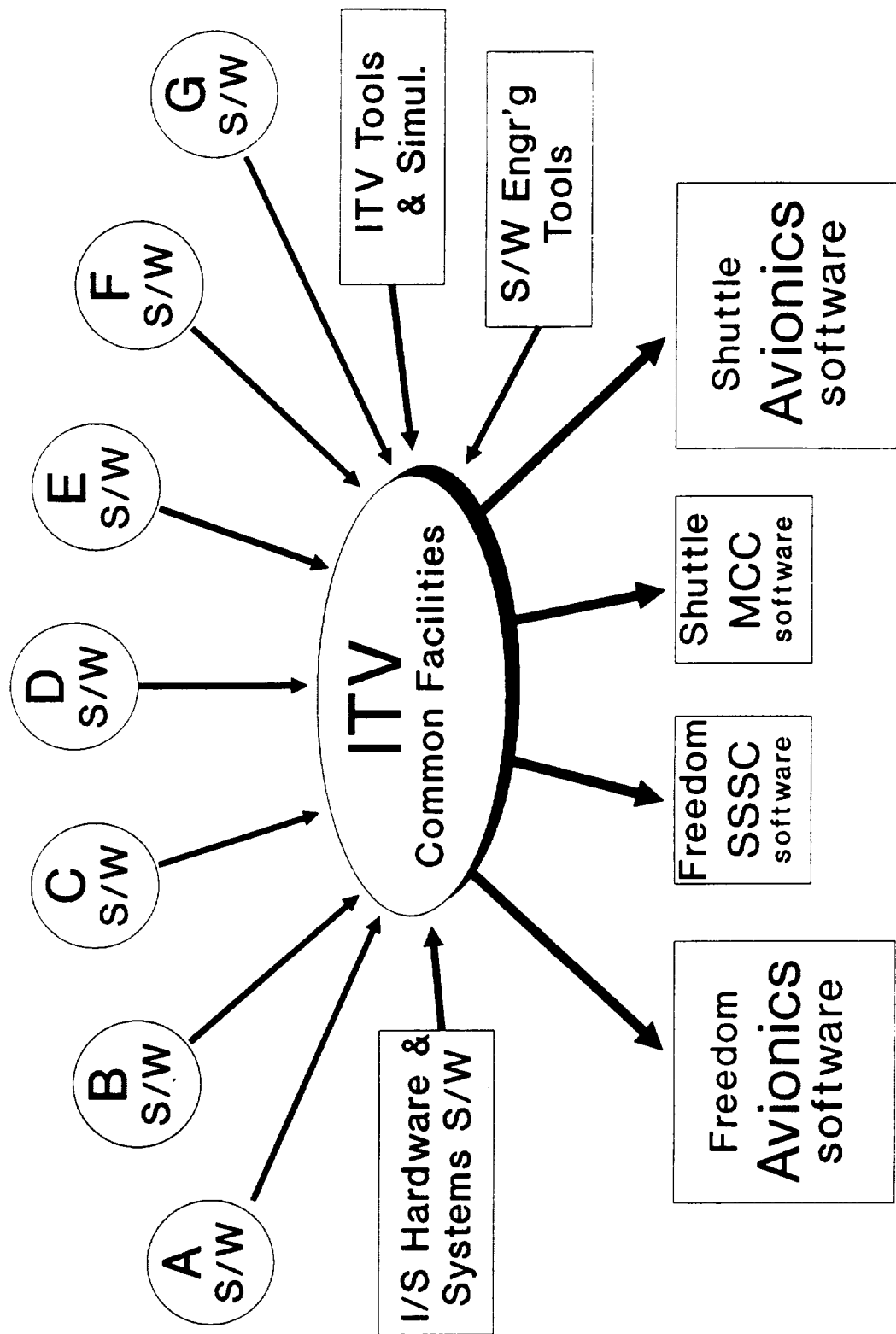
Need Dates ▲ ▲ ▲

SSFP STS

Lunar/Mars

• (Technology Phases)

SPACE TRANSPORTATION AVIONICS TECHNOLOGY SYMPOSIUM  
ADVANCED SOFTWARE INTEGRATION



# Major Objectives

- Maintaining reliability in increasingly complex software and information systems (contrasts in STS and SSFP avionics).
- Enabling evolution (functionality, technology, connectiveness) in systems which are now “never-ending”.
- Managing increasingly distributed work-packets and efforts in the development of applications software for the advanced systems.
- Reuse and commonality (across systems and programs) both an operations efficiency (training, management, etc.) and as a productivity item.

# Key Contacts & Facilities

## Contacts:

John R. Garman/JSC(FA)  
Ed Chevers/JSC(FR)  
Rick Coblentz/JSC(FR)  
Jack Seyl/JSC(FS)  
Charles McKay/UHCL (JSC)

## Facilities:

Information Systems Technology Lab (ISTL)/JSC(FA)  
Avionics Development Lab (SSFP/WP2)/JSC(FR)  
Software Development Facility (NSTS)/JSC(FR)  
Support Software Environment Development Facility  
(SSEDF)/JSC(FR)  
Mission Systems ITV Facility/JSC(FS)



## Major Milestones (1990-1995)

- Operation of STS SAIL and SPF (1976)
- SSE Baseline for SSFP (1990)
- Avionics Integration and ITV baseline for SSFP (1990)
- Mission Systems ITV (MSITV) Facility Design (1990)
- ADF and MSITV FOC (1992)
- Shuttle and SSFP ITV commonality (1997)

# Technology Issues

- Containment of growing drivers: complexity, connectivity, security, and architectures
- Standardization of I/S “layers” - industry standards
- Virtual target environments (exact simulation of target platform allowing diagnostics)
- “Project Object Database” - the database and management technologies involved in creating a single unambiguous image of the entire distributed software system
- Integration of heterogeneous products designed against common standards (both the host and target domains)
- Software Lifecycles modeled against evolutionary development and maintenance (vs. waterfall)

# Candidate Programs

- NSTS Avionics Flight Software
- NSTS Mission Control Center Upgrade
- NSTS Other I/S
- SSFP Data Management System (avionics)
- SSFP Mission Control and Trainers
- SSFP Other I/S
- Advanced Programs (Lunar/Mars)

# Major Accomplishments/“Inabilities”

- ✓ Major Accomplishments
  - Establishment of RICIS
  - Establishment of SSE development effort
  - Baselining of commonality in applications tools and UI for SSFP
  - Industry evolution toward standardization of I/S layers
- ✓ Major “Inabilities”
  - Duplication of effort across Programs/Projects
  - Proliferation of mission supporting software
  - Inability to fully utilize COTS
  - Inability to upgrade existing capabilities

# SIGNIFICANT MILESTONES

82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

\* T 8 R

GSFC SEL, JSC RICIS, CMU SEI

Adv. Devel. \*

SSFP SSE, JSC ISTL

DOT & E\*

ADF, MSITV (SAIL-2?)

### Projected Level 6 Tech. Maturity

## Need Dates

**SSP STS'**

## Lunar/Mars

\* (Technology Phases)



## FLIGHT ELEMENTS PANEL

PRECEDING PAGE BLANK NOT FILMED

